

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1-31. (Canceled)
- 1 32. (Currently amended): A method of transforming data, the method
- 2 comprising:
- 3 positioning a definition pointer to point at a first compound transform definition
- 4 within a transform process definition;
- 5 invoking a first parallel processing thread to read the pointed at first compound
- 6 transform definition;
- 7 searching data to be transformed for a data element to be transformed, the search
- 8 being responsive to the first compound transform definition;
- 9 calling a dynamic function defined in the first compound transform definition, the
- 10 dynamic function located elsewhere from the definition pointer position;
- 11 transforming any found data element into an output data file, responsive to the
- 12 first compound transform definition and called dynamic function, a data structure of the output
- 13 data file being responsive to a data structure of the transform process definition;
- 14 positioning a definition pointer to point at a second compound transform
- 15 definition within the transform process definition;
- 16 invoking a second parallel processing thread to read the pointed at second
- 17 compound transform definition;
- 18 searching data to be transformed for another data element to be transformed, the
- 19 search being responsive to the second compound transform definition; and
- 20 transforming any found data element into the output data file, responsive to the
- 21 second compound transform definition, the data structure of the output data file being responsive
- 22 to the data structure of the transform process definition;

23 ~~wherein the read first compound transform definition includes a translation~~
24 ~~codeset parameter enabling the transforming to include a call to one of a function or a lookup~~
25 ~~table located in the first compound transform definition.~~

1 33. (Previously presented): The method of claim 32, further including
2 determining a type of the read first compound transform definition and, if the first compound
3 transform definition is not a simple transform definition type, recursively calling the method of
4 claim 32.

1 34. (Original): The method of claim 32, further including determining if all
2 sub-definitions of a compound transform definition have been processed.

1 35. (Original): The method of claim 32, wherein the method of transforming
2 data includes nesting of a data element.

1 36. (Previously presented): The method of claim 32, further including, if no
2 data element is found in either step of searching data to be transformed, adding an output data
3 element to the output data file responsive to the read first compound transform definition, the
4 data to be transformed having no contribution to the output data element.

1 37. (Previously presented): The method of claim 32, wherein the read first
2 compound transform definition includes a value parameter configured to specify a value for
3 inclusion in the output data file.

1 38. (Original): The method of claim 32, wherein the data element is a
2 compound data element and the read transform definition includes a source record parameter
3 configured to specify the compound data element.

1 39. (Previously presented): The method of claim 32, wherein the read first
2 compound transform definition is in a meta-language format.

1 40. (Original): The method of claim 32, wherein the data to be transformed
2 data is in a meta-language data format.

1 41. (Previously presented): The method of claim 32, wherein the read first
2 compound transform definition includes a transform element having an output field name and a
3 source field parameter.

1 42. (Previously presented): The method of claim 32, wherein the read first
2 compound transform definition includes a value parameter configured to populate a field in the
3 output data file.

1 43. (Canceled)

1 44. (Currently amended): A method of transforming data, the method
2 comprising:
3 positioning a definition pointer to point at a first compound transform definition,
4 the first compound transform definition being within a transform process definition;
5 invoking a first parallel processing thread to read the pointed at first compound
6 transform definition and sub-definitions of the first compound transform definition;
7 positioning a first payload pointer to point at a data element to be transformed, the
8 positioning being responsive to a data structure of the first compound transform definition;
9 calling a dynamic function defined in the first compound transform definition, the
10 dynamic function located elsewhere from the definition pointer position;
11 transforming the data element into an output data file, responsive to the read first
12 compound transform definition and called dynamic function;
13 positioning the definition pointer to point at a second compound transform
14 definition, the second compound transform definition being within the transform process
15 definition;

16 invoking a second parallel processing thread to read the pointed at second
17 compound transform definition and sub-definitions of the second compound transform
18 definition;
19 positioning a second payload pointer to point at a data element to be transformed,
20 the positioning being responsive to a data structure of the second compound transform definition;
21 and
22 transforming the second data element into the output data file, responsive to the
23 read second compound transform definition.

1 45. (Previously presented): The method of claim 44, further including
2 determining a type of the read first compound transform definition and, if the read first
3 compound transform definition is not a simple transform definition type, recursively calling the
4 method of claim 44.

1 46. (Previously presented): The method of claim 44, further including
2 determining a type of the read first compound transform definition and, if the read first
3 compound transform definition is not a simple transform definition type, recursively calling the
4 method of claim 44, wherein the recursive call is responsive to the data structure of the transform
5 process definition.

1 47. (Previously presented): The method of claim 44, further including
2 determining a type of the read first compound transform definition, if the read transform
3 definition is not a simple transform definition type recursively calling the method of claim 44,
4 and determining if all sub-elements of a compound element have been transformed.

1 48. (Original): The method of claim 44, further including determining if all
2 sub-elements of a compound element have been transformed and, if the determination returns a
3 value of YES, returning to a calling process.

1 49. (Canceled)

1 50. (Original): The method of claim 44, wherein the method of transforming
2 data includes un-nesting of the data element to be transformed.

1 51. (Previously presented): The method of claim 44, wherein the read first
2 compound transform definition includes a source field parameter configured to specify the data
3 element.

1 52. (Previously presented): The method of claim 44, wherein the read first
2 compound transform definition includes a source record parameter configured to specify the
3 compound data element.

1 53-54. (Canceled)

1 55. (Original): The method of claim 44, further including a step of combining
2 the data element with the transform process definition prior to transforming the data element to
3 output data.

1 56. (Original): The method of claim 44, wherein the transform process
2 definition includes a tree data structure.

1 57-58. (Canceled)

1 59. (Previously presented): A computer readable storage media having
2 embodied thereon data, the data comprising:
3 computer instructions configured to position a definition pointer to point at a first
4 compound transform definition, the first compound transform definition being within a transform
5 process definition;
6 computer instructions configured to invoke a first parallel processing thread to
7 read the pointed at first compound transform definition and sub-definitions of the first compound
8 transform definition;

9 computer instructions configured to increment position a first payload pointer;
10 within the data to be transformed, to point at a data element to be transformed, the
11 incrementation positioning being responsive to a data structure of the pointed-at-first compound
12 transform definition;

13 computer instructions configured to call a dynamic function defined in the first
14 compound transform definition, the dynamic function located elsewhere from the definition
15 pointer position;

16 computer instructions configured to transform any found the data element into an
17 output data file, responsive to the read first compound transform definition and called dynamic
18 function;

19 computer instructions configured to increment position a second payload pointer;
20 within the data to be transformed, to point at a data element to be transformed, the
21 incrementation positioning being responsive to a data structure of the pointed-at-second compound
22 transform definition;

23 computer instructions configured to invoke a second parallel processing thread to
24 read the pointed at second compound transform definition and sub-definitions of the second
25 compound transform definition; and

26 computer instructions configured to transform the second data element into the
27 output data file, responsive to the read second compound transform definition.

1 60. (Original): The computer readable media of claim 59, wherein the data
2 further comprises computer instructions configured to employ recursion to transform a
3 compound data element within the data to be transformed.

1 61-64. (Canceled)

1 65. (Currently amended): An application system comprising:
2 a computing device;
3 means for positioning a definition pointer to point at a first compound transform
4 definition within a transform process definition;
5 means for invoking a first parallel processing thread to read the first compound
6 transform definition by the computing device;
7 means for calling a dynamic function defined in the first compound transform
8 definition, the dynamic function located elsewhere from the definition pointer position;
9 means for positioning the definition pointer to point at a second compound
10 transform definition within the transform process definition;
11 means for invoking a second parallel processing thread to read the second
12 compound transform definition by the computing device;
13 means for positioning a payload pointer to point to a first data element, the first
14 data element being a member of a plurality of data elements within data to be transformed; and
15 means for generating an output data file using the first data element and the first
16 and second compound transform definitions;
17 wherein the means for positioning the definition pointer and the means for
18 positioning the payload pointer are enabled to be invoked concurrently.

1 66. (Original): The application system of claim 65, further including means
2 for selecting the transform process definition from a set of transform process definitions,
3 responsive to data associated with the data to be transformed.

1 67. (Original): The application system of claim 65, wherein a second data
2 element has no contribution to output data generated using the transform process definition, the
3 second data element being a member of the plurality of data elements.

1 68. (Previously presented): The application system of claim 65, further
2 including means for adding data to the output data file, the added data being configured

Appl. No. 10/661,167
Amdt. dated May 24, 2010
Reply to Office Action of March 23, 2010

PATENT

- 3 responsive to the transform process definition and having no contribution from the data to be transformed.
- 4